2.25 Verification Results for Signal Integration

No major errors were found in the Signal Integration FE for ALARM 3.0; however some minor discrepancies between the code and the input guide in the User's Manual were found. The overall code quality is good, but a few improvements were recommended.

The table listed below summarizes the desk-checking and software testing verification activities for each subroutine in the Signal Integration Functional Element. The two results columns contain checks if no discrepancies were found. Where discrepancies were found, the desk check results column contains references to discrepancies listed in table 2.25-4, while the test case results column lists the number of the relevant test case in table 2.25-6. More detailed information on the results is recorded in these tables.

Table 2.25-1 Verification Results Summary

DESIGN ELEMENT	CODE LOCATION	DESK CHECK RESULT	TEST CASE ID	TEST CASE RESULT
25-1 Integration Gain	THRESH 265-266 GETRCS 214 and 225	✓	25-3,4,5	✓
25-2 Detectability Factor with No Fluctuations	THRESH 101-131 and 239-253	D1	25-1 and 25-2	✓
25-3 Inclusion of Fluctuation Loss	THRESH 265-266	✓	25-3,4	✓
Input	RDRINP	✓	25-6	✓
Initialization	RDRINT 456-460	✓	25-5	✓
Echo Inputs	RDRPRT	✓	25-6	✓
Error Check	RDRERR 156-165 181-198 207-212 and 429-441	D2	25-6,7,8	25-8

2.25.1 Overview

Signal integration is the summation of several signal and noise pulses for the purpose of improving the detectability of target signals. Integration gain is the resulting improvement factor in the signal-to-noise ratio. The Signal Integration FE in ALARM 3.0 simulates the noncoherent integration performed in target detection for radar systems with some common types of detectors and integrators.

The ALARM 3.0 implementation of Signal Integration is done primarily in subroutine THRESH. THRESH and other subroutines used for this FE are described in table 2.25-2. None of these routines is dedicated to integration, they also implement other FEs (especially Signature Fluctuations and Detection Threshold). Only the portions applicable to integration were verified.

Table 2.25-2 Subroutine Descriptions

MODULE NAME	DESCRIPTION
THRESH	Calculates integration gain and fluctuations loss (or the detection threshold of the radar) given probability of detection and false alarm
GETRCS	Extracts and interpolates the RCS of the target and converts integration gain from db to absolute
RDRERR	Checks for legality of user input data for radar parameters
RDRINP	Reads radar data
RDRINT	Performs initial processing on user inputs for radar parameters
RDRPRT	Prints user inputs for radar parameters

2.25.2 Verification Design Elements

Design elements defined for the signal integration FE are listed in table 2.25-1. A design element is an algorithm that represents a specific component of the FE design. The first three design elements are fully described in Section 2.25.2 of the ASP II for ALARM 3.0.

Table 2.25-3 Integration Design Elements

SUBROUTINE	DESIGN ELEMENT	DESCRIPTION
THRESH and GETRCS	25-1 Integration Gain	Calculate the improvement in detectability due to integrating N pulses
THRESH	25-2 Detectability Factor with No Fluctuations	Calculate the detectability factor, $\bar{D}_o(N)$, for nonfluctuating target
THRESH	25-3 Inclusion of Fluctuation Loss	Calculate the detectability factor, $D_o(i)$, including fluctuation.
RDRINP	Input	Read user inputs in DATARADR
RDRINT	Initialization of NPULSS	Initialize NPULSS for all values of IPRF
RDRERR	Error Checks	Check user inputs in DATARADR to insure they are within appropriate limits.
RDRPRT	Echo Inputs	Print DATARADR input values

2.25.3 Desk Checking Activities and Results

The code implementing this FE was manually examined using the procedures described in Section 1.1 of this report. Discrepancies discovered are described in the following tables.

Table 2.25-4 Code Discrepancies

DESIGN ELEMENT	DESK CHECK RESULT
25-2 Detectability Factor with no Fluctuations	D1. Desk Checking found potential problems when entering NPULSE values. If there is an ideal or uniform weight integrator with no signal variation , then the ordinary number of pulses integrated can be used. In other cases, user must enter equivalent number of pulses integrated (table 2.2 in Blake).
Error Checks	D2. According to Blake [A.1-4], P_d and P_{fa} must be more strictly bounded to use these algorithms. ALARM limits both to [0,1], but bounds should be $0.1 \le P_d \le 0.9$ and $10^{-12} \le P_{fa} \le 10^{-4}$.

Except as noted in table 2.25-5 below, overall code quality and internal documentation were evaluated as good. Subroutine I/O and logical flow were found to match the ASP II descriptions.

Table 2.25-5 Code Quality and Internal Documentation Results

SUBROUTINE	CODE QUALITY	INTERNAL DOCUMENTATION
THRESH	OK	The comment for HSUB1 and HSUBN at lines 125-128 should mention the assumptions that make H_N=NPULSE. These conditions are mentioned in the comment at lines 93-99 about ETA, but the comments do not connect these assumptions with using NPULSE for H_N.
GETRCS	OK for integration (Please refer to table 2.4.5 in Section 2.4 of this report)	The header lacks description of purpose, some inputs, and outputs.
RDRINT	OK	Header lacks definition of some input and output variables.
RDRINP RDRERR RDRPRT	These contain code dealing with unused variables (STCMXR, STCMNR, and STCATN). In addition, there is a problem with units in the print-out of these variables; the same numbers input in meters get printed out as kilometers.	OK

2.25.4 Software Test Cases

The software testing was performed by running the entire ALARM model in debug mode. For these tests, ALARM was run in contour mode using the input data files for Sample 13 that were delivered with ALARM code. Any data modifications are listed in each test description.

Subroutine THRESH implements portions of several FEs. Only the portions for integration (101-131 and 239-269) were tested. In GETRCS only lines 215 and 225 implement the integration FE. These lines convert the integration gain from db to absolute. The tests 25-1 through 25-5 were designed to test design elements 25-1 through 25-3.

The remaining test cases (25-6 through 25-8) addressed the user input and error checking routines. RDRINP reads the input data for this FE, RDRERR checks those inputs for errors, and RDRPRT echoes them.

Table 2.25-6 Software Test Cases for Signal Integration

TEST CASE ID	TEST CASE DESCRIPTION
25-1	OBJECTIVE: Check that correct value of asymptotic efficiency of envelope detector is used for both detector types in subroutine THRESH. PROCEDURE: 1. Set ISQLAW = 0 in the Sample 13 input file. 2. Run ALARM, using Sample 13 as the input file. 3. Set a breakpoint in subroutine THRESH. 4. At line 101 observe the value of ISQLAW. 5. Observe the value of ETA and compare with equation (2.25-6) of ASP II. 6. Repeat steps 1-5 with ISQLAW = 1. VERIFY: ETA is correctly initialized. RESULT: OK
25-2	OBJECTIVE: Check computation of detectability factors in subroutine THRESH. PROCEDURE: 1. Run ALARM, using Sample 13 as the input file. 2. Set a breakpoint in subroutine THRESH. 3. Deposit the following values: ISQLAW = 0 PSUBFA = 10^{-8} PSUBD = 0.8 4. At line 103 note the value of ETA. 5. At line 247 note the value of ABSD1. 6. At line 249 note the value of ABSDN. 7. Independently calculate ABSD1 and ABSDN using equation (2.25-2) of ASP II; compare to values in steps 5 and 6. 8. Repeat steps 1-7 with ISQLAW = 1. VERIFY: ALARM values match independent calculations. RESULT: OK

Table 2.25-6 Software Test Cases for Signal Integration

TEST CASE ID	TEST CASE DESCRIPTION
25-3	OBJECTIVE: Check final computation of integration gain in THRESH. PROCEDURE: 1. Run ALARM, using Sample 13 as the input file. 2. Set a breakpoint in subroutine THRESH. 3. Make the following deposits: ISQLAW = 1.0 PSUBD = 0.8 PSUBFA = 10^{-8} 4. At lines 233 and 235 observe the values of ABSLF1 and ABSLFN. 5. At lines 265 and 266 observe the values of CONTOR and DBGAIN. 6. Independently calculate CONTOR and DBGAIN using equations (2.25-8) and (2.25-1) of ASP II; compare to values in steps 4 and 5. VERIFY: ALARM values match independent calculations. RESULT: OK
25-4	OBJECTIVE: Check parameter passing and units conversion for pulsed (MTI) radars in THRESH and GETRCS. PROCEDURE: 1. Run ALARM, using Sample 13 as input. 2. Set a breakpoint in subroutines GETRCS and THRESH. 3. Observe the values of ABSLF1 and ABSLFN at lines 214-216 and the values of ABSD1 and ABSDN at lines 247-249 in THRESH. 4. At line 266 of THRESH note the value of DBGAIN. 5. Independently calculate DBGAIN combining values in step 3 and then converting to db; compare to value in step 4. 6. Observe the value of GANINT at line 225 of GETRCS. 9. Independently calculate GANINT using values in step 3; compare to value in step 6. VERIFY: ALARM values match independent calculations. RESULT: OK

Table 2.25-6 Software Test Cases for Signal Integration

TEST CASE ID	TEST CASE DESCRIPTION
25-5	OBJECTIVE: Check parameter passing and units conversion for pulse-doppler radars in THRESH and GETRCS. PROCEDURE: 1. Run ALARM, using Sample 8 as the input file. 2. Set breakpoints in GETRCS and THRESH. 3. At line 210 of GETRCS note the value of NPULSS for each PRF. 4. At lines 214-216 and 247-249 of THRESH note the values of ABSLF1, ABSLFN, ABSD1, and ABSDN. 5. Return to GETRCS and observe the value of GNINTS(1). 6. Independently calculate GNINTS(1)using values in step 4; then compare to value in step 5. 7. Repeat steps 4-6 for each PRF. VERIFY: 1. NPULSS equals NPULSE for each PRF. 2. ALARM values match independent calculations. RESULT: OK
25-6	OBJECTIVE: Check for correct reading, printing, and error checking of inputs. PROCEDURE: 1. Run ALARM, using Sample 8 as the input file. 2. Examine the printed output for DATARADR and compare with independent assessment of inputs. VERIFY: 1. Printed ALARM output matches independent assessment of inputs. 2. No errors are generated. RESULT: OK
25-7	OBJECTIVE: Check error-handling in RDRERR. PROCEDURE: 1. Revise the Sample 8 input file to include the following errors: IPRRDR = 2 IRADAR = -5 ISQLAW = 3 NPRFS = 5 NPULSE = -10 PSUBFA = 2.0 PSUBD = -1.5 Leave other records unchanged. 2. Independently predict error messages that should be generated. 3. Run ALARM using Sample 8 with revised radar data as input. 4. Compare ALARM-generated errors to independent predictions. VERIFY: ALARM errors match predicted errors. RESULT: OK

Table 2.25-6 Software Test Cases for Signal Integration

TEST CASE ID	TEST CASE DESCRIPTION
25-8	OBJECTIVE: Check error-handling for END-OF-FILE in RDRERR. PROCEDURE: 1. Revise the Sample 13 input file by deleting all records from the last record in DATARADR to the end of file. 2. Run ALARM using the revised Sample 13 input file. 3. Break in subroutine RDRINP. 4. Observe the next executable line after line 316. VERIFY: Subroutine RDSTAT is called when an END-OF-FILE is reached. RESULT: RDSTAT is not called if an END-OF-FILE is reached. No other input is read and RDRINP returns to the calling program.

2.25.5 Conclusions and Recommendations

Code Discrepancies

No major errors were found. Errors in checking theoretical limits on P_{fa} and P_d were found in RDRERR; these errors are due to using referenced algorithms without maintaining the limits specified by the reference. This should be corrected by using the bounds described in table 2.25-4. Error-handling for premature end-of-file should also be modified. It seems that subroutine RDSTAT is meant to be called for this error, but it is not.

Code Quality and Internal Documentation

Code quality is generally good; although there are some unused input variables with minor discrepancies between the printed output from RDRPRT and the ALARM input guide. Internal documentation is adequate, but should be improved to provide a consistent set of information in the subroutine headers. Additional comments are recommended to explain the assumptions that make H_N = NPULSE.

External Documentation

The external documentation is somewhat inadequate. The Input Guide in the User's Manual gives incorrect bounds on PSUBD and PSUBFA. According to Blake [A.1-4], they should be $0.1 \leq PSUBD \leq 0.9$ and $10^{-12} \leq PSUBFA \leq 10^{-4}$. In addition, the Input Guide should warn users that the value entered for NPULSE should be the equivalent number of pulses integrated and refer to table 2.2 in Blake [A.1-4]. (The Analyst's Manual warns users to carefully define the number of pulses integrated, but it does not give specifics or mention equivalent number of pulses integrated.) Also, the Input Guide should refer to parameter names (not specific values) for array dimensions.